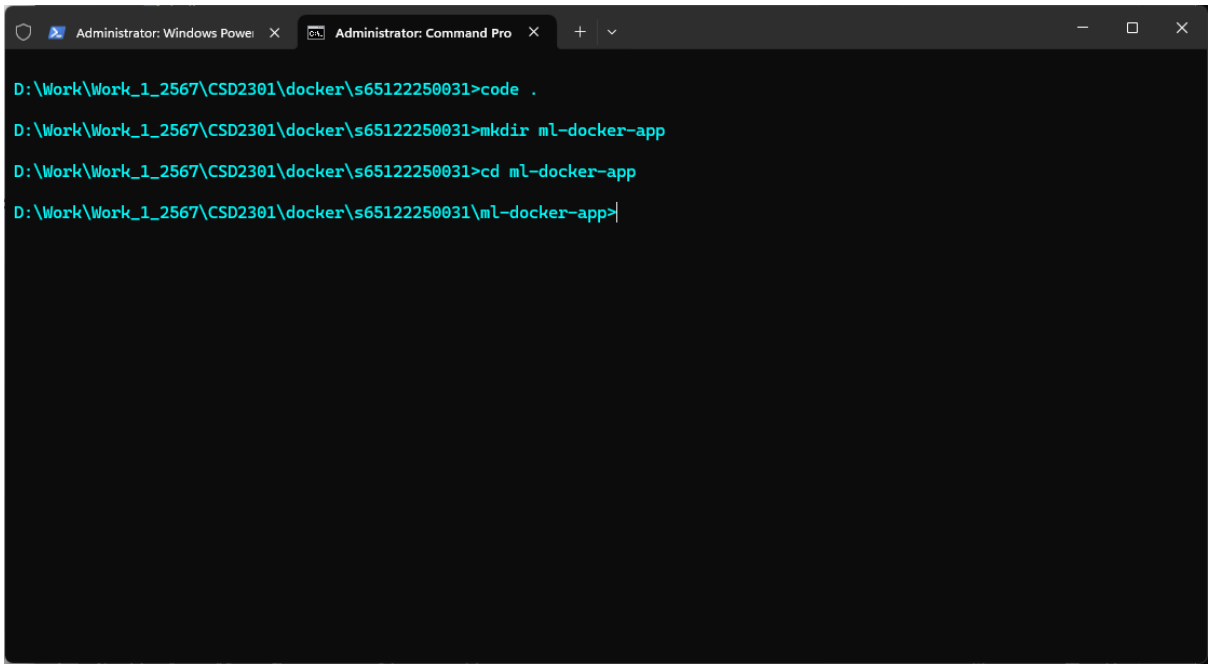
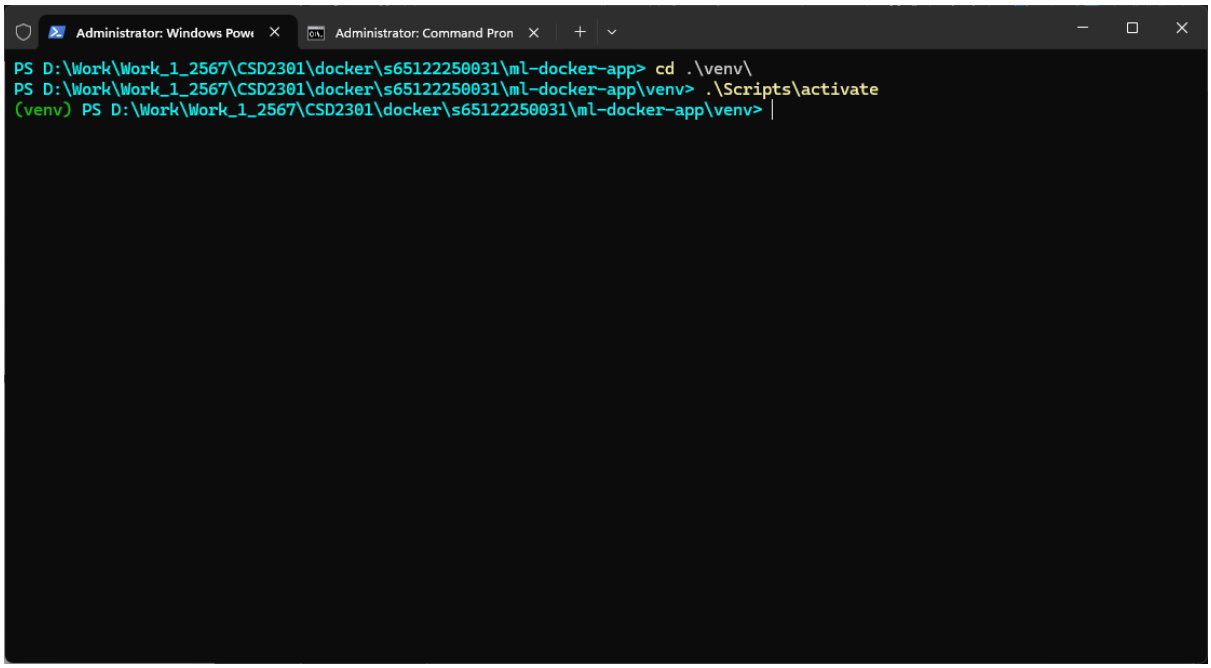
Lab ML and Docker

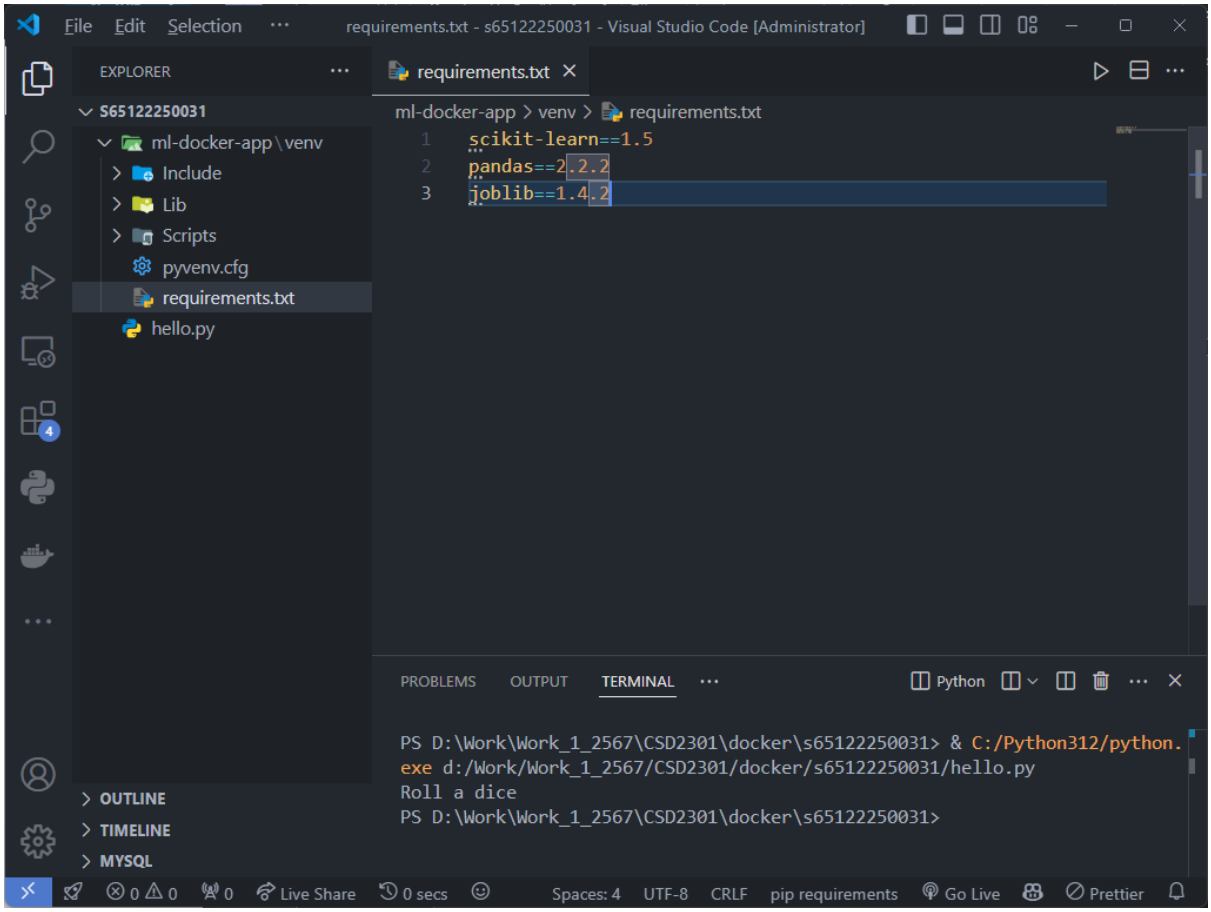
1. Docker Installed
2. Run Power Shell as Administrator
3. Project directory



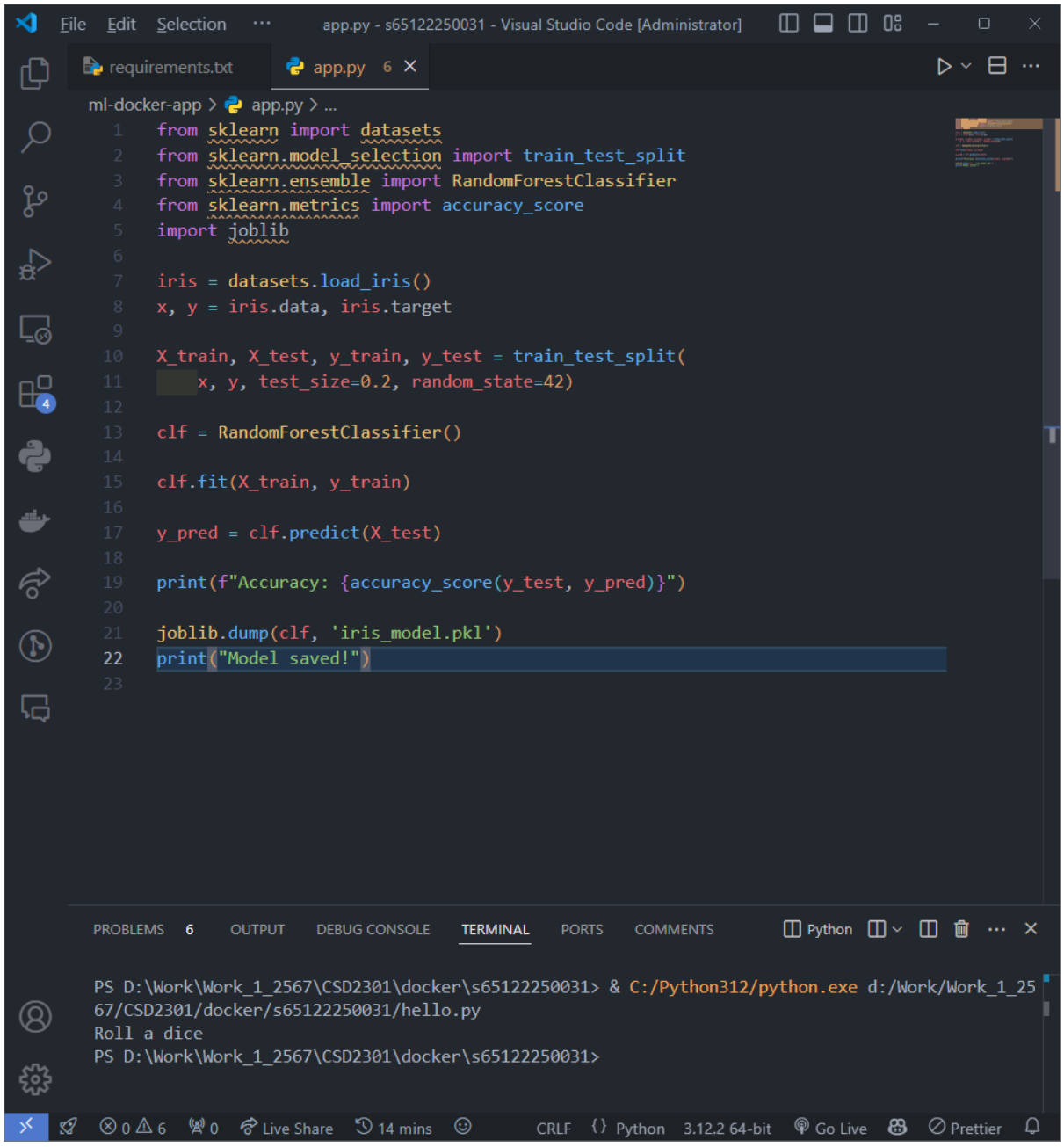
4. Set Up a Python Virtual Environment



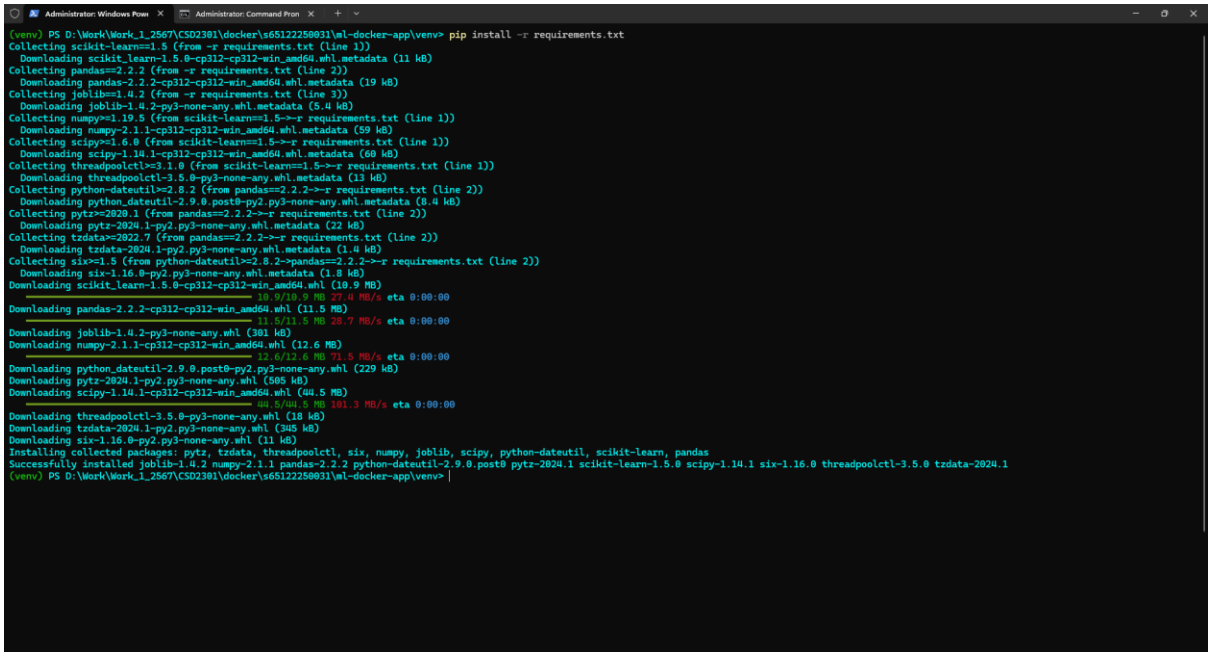
5. Requirement.txt File



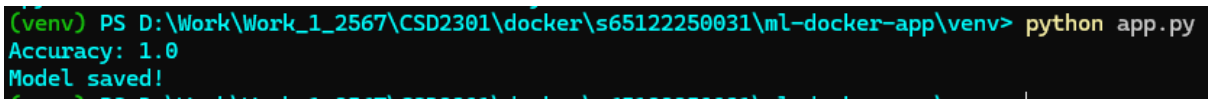
6. Script


File Edit Selection
app.py - s65122250031 - Visual Studio Code [Administrator]
requirements.txt
app.py 6
ml-docker-app > app.py > ...
1 from sklearn import datasets
2 from sklearn.model_selection import train_test_split
3 from sklearn.ensemble import RandomForestClassifier
4 from sklearn.metrics import accuracy_score
5 import joblib
6
7 iris = datasets.load_iris()
8 x, y = iris.data, iris.target
9
10 X_train, X_test, y_train, y_test = train_test_split(
11 x, y, test_size=0.2, random_state=42)
12
13 clf = RandomForestClassifier()
14
15 clf.fit(X_train, y_train)
16
17 y_pred = clf.predict(X_test)
18
19 print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
20
21 joblib.dump(clf, 'iris_model.pkl')
22 print("Model saved!")
23
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python
PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031> & C:/Python312/python.exe d:/Work/Work_1_2567/CSD2301/docker/s65122250031/hello.py
Roll a dice
PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031>
0 6 0 Live Share 14 mins CRLF Python 3.12.2 64-bit Go Live Prettier

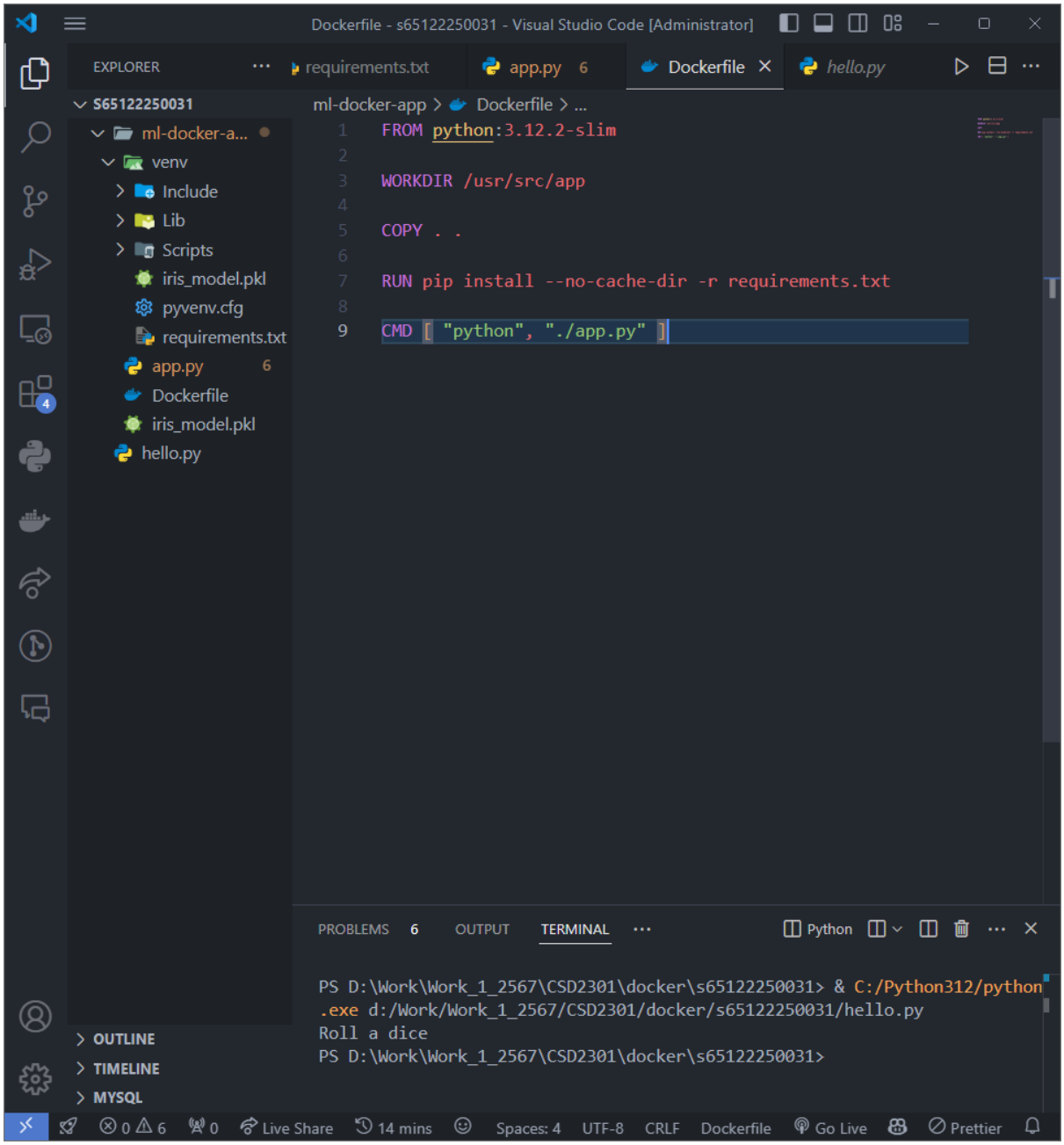
7. Install the Dependencies



8. Run application



9. Dockerfile



10. Docker Image

```
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker build -f Dockerfile -t my-ml-app .
[+] Building 36.5s (9/9) FINISHED                                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 184B                                                                                0.0s
 => [internal] load metadata for docker.io/library/python:3.12.2-slim                                               1.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [1/4] FROM docker.io/library/python:3.12.2-slim@sha256:5dc6f84b5e97bfb0c90abfb7c55f3cacc2cb6687c8f920b64a833a   0.0s
 => [internal] load build context                                                                                   9.9s
 => => transferring context: 341.25MB                                                                               9.9s
 => CACHED [2/4] WORKDIR /usr/src/app                                                                               0.0s
 => [3/4] COPY . .                                                                                                  2.4s
 => [4/4] RUN pip install --no-cache-dir -r requirements.txt                                                       20.9s
 => exporting to image                                                                                              2.2s
 => => exporting layers                                                                                             2.1s
 => => writing image sha256:17a33b4d3362d4df1e0ed976f4cacc458b153fd105005c9656e81526017b2b93                        0.0s
 => => naming to docker.io/library/my-ml-app                                                                        0.0s
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker build -t ml-docker-app .
[+] Building 1.6s (9/9) FINISHED                                                                     docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 184B                                                                                0.0s
 => [internal] load metadata for docker.io/library/python:3.12.2-slim                                               1.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [1/4] FROM docker.io/library/python:3.12.2-slim@sha256:5dc6f84b5e97bfb0c90abfb7c55f3cacc2cb6687c8f920b64a833a   0.0s
 => [internal] load build context                                                                                   0.5s
 => => transferring context: 947.84kB                                                                               0.5s
 => CACHED [2/4] WORKDIR /usr/src/app                                                                               0.0s
 => CACHED [3/4] COPY . .                                                                                           0.0s
 => CACHED [4/4] RUN pip install --no-cache-dir -r requirements.txt                                                 0.0s
 => exporting to image                                                                                              0.0s
 => => exporting layers                                                                                             0.0s
 => => writing image sha256:17a33b4d3362d4df1e0ed976f4cacc458b153fd105005c9656e81526017b2b93                        0.0s
 => => naming to docker.io/library/ml-docker-app                                                                    0.0s
```

11. Run Docker container

```
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker image

Usage:  docker image COMMAND

Manage images

Commands:
  build       Build an image from a Dockerfile
  history     Show the history of an image
  import      Import the contents from a tarball to create a filesystem image
  inspect     Display detailed information on one or more images
  load        Load an image from a tar archive or STDIN
  ls          List images
  prune       Remove unused images
  pull        Download an image from a registry
  push        Upload an image to a registry
  rm          Remove one or more images
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

Run 'docker image COMMAND --help' for more information on a command.
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker run -it --rm my-ml-app:latest
Accuracy: 1.0
Model saved!
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv>
```

12. Tag & Push the container to dockerhub